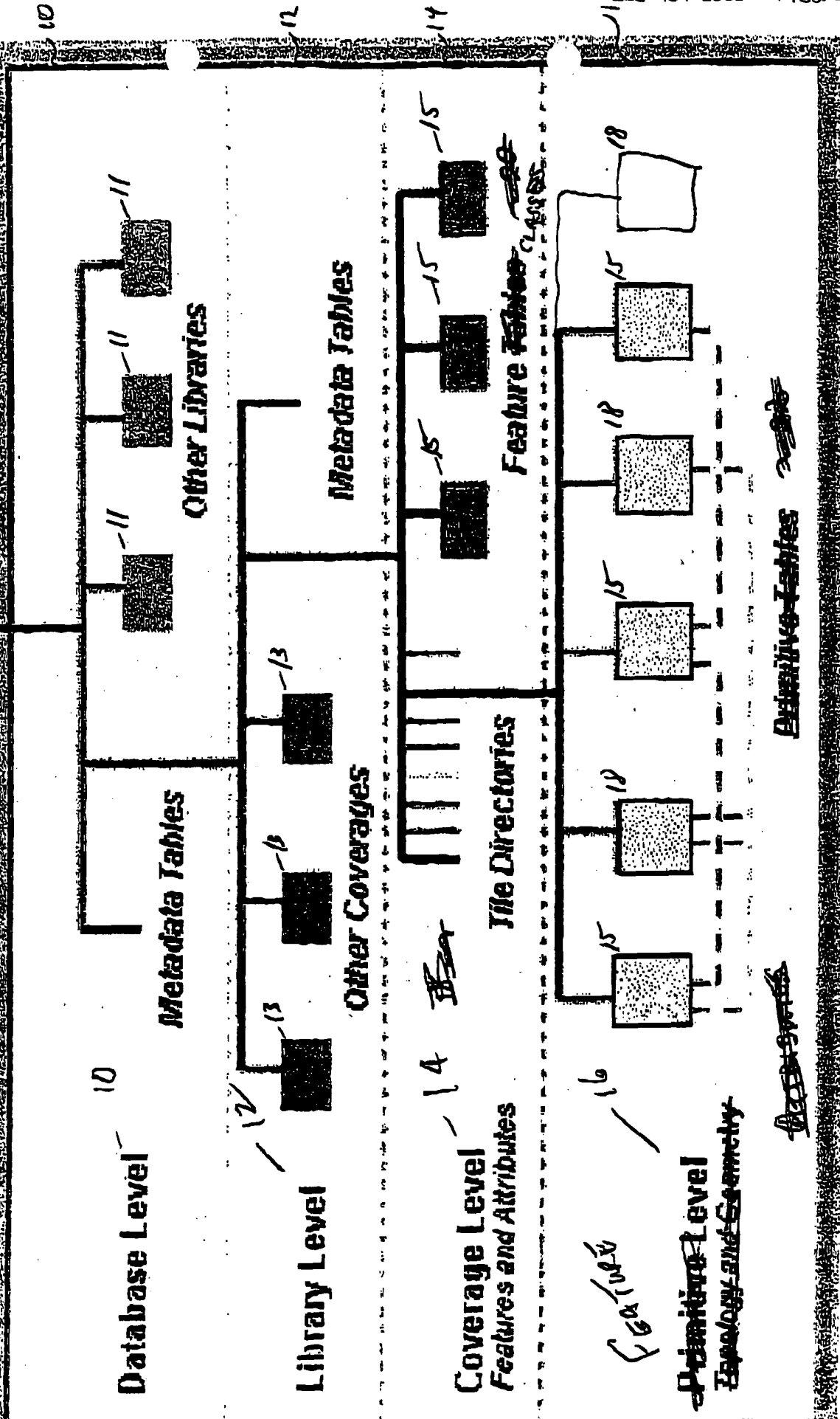


VPEE DATA STRUCTURE



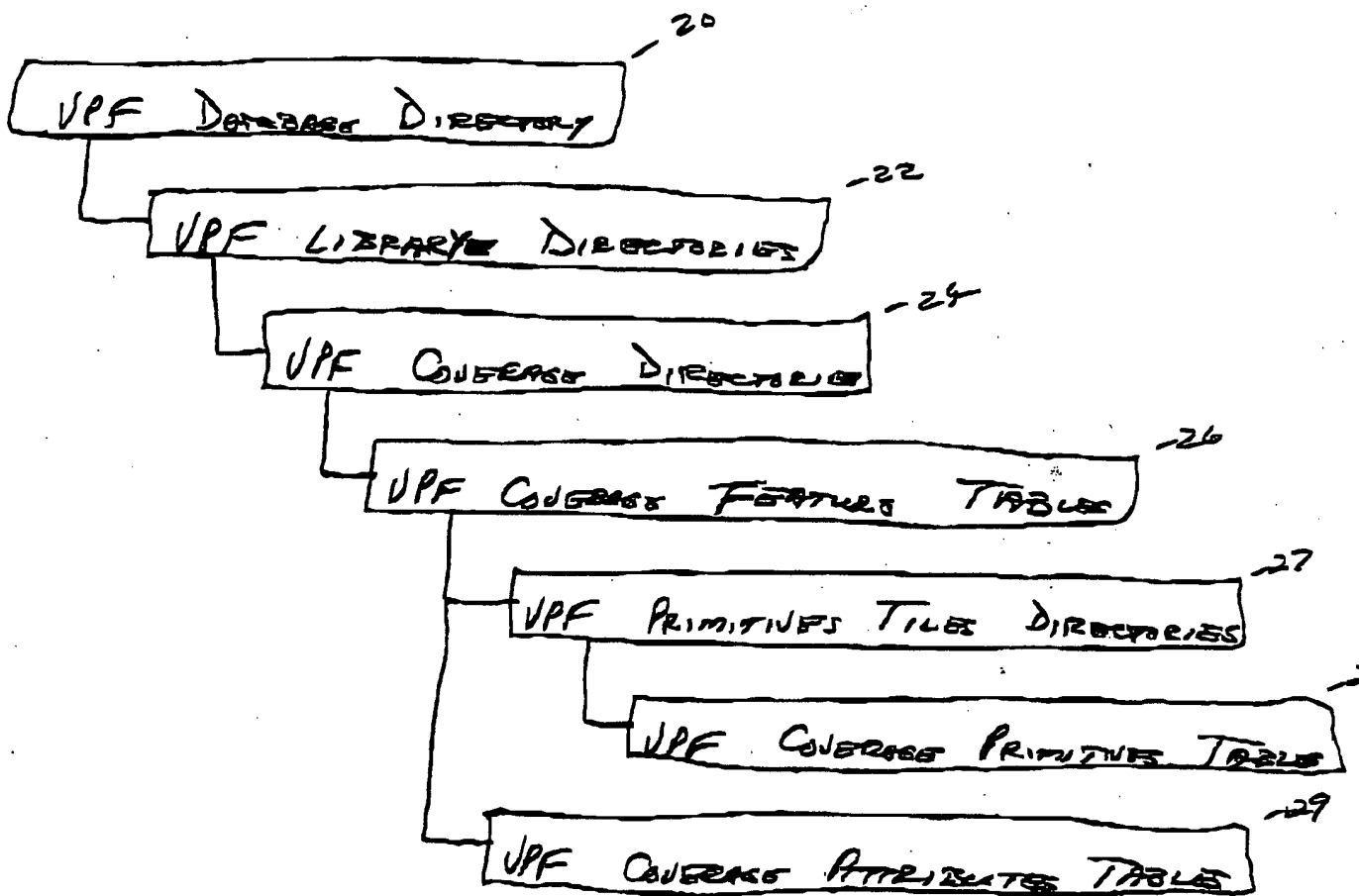


FIG 2

Prior Art

202 434 1501 P. 37/49
with these definitions, winged-edge topology
partition of neighboring nodes, edges and

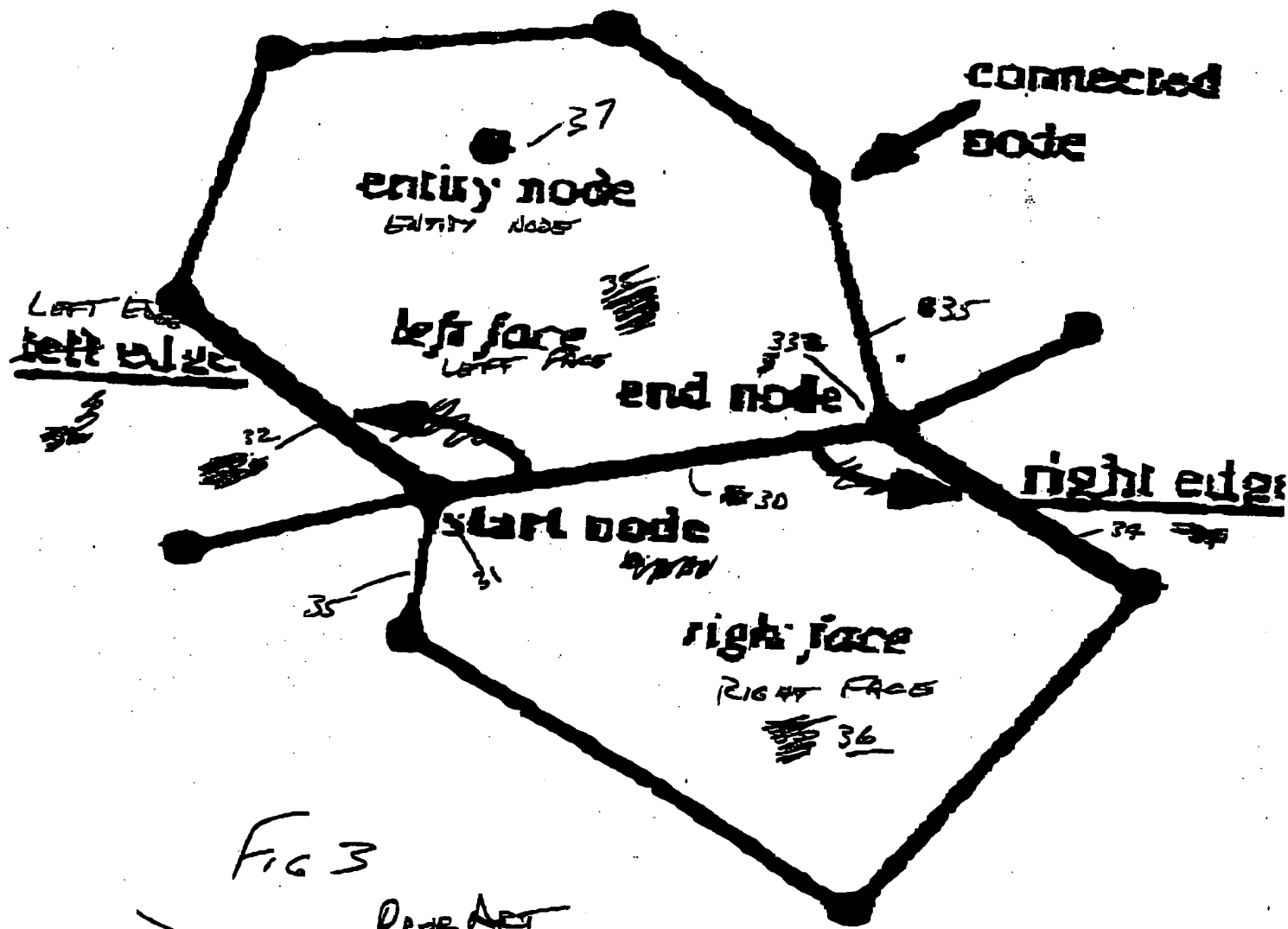


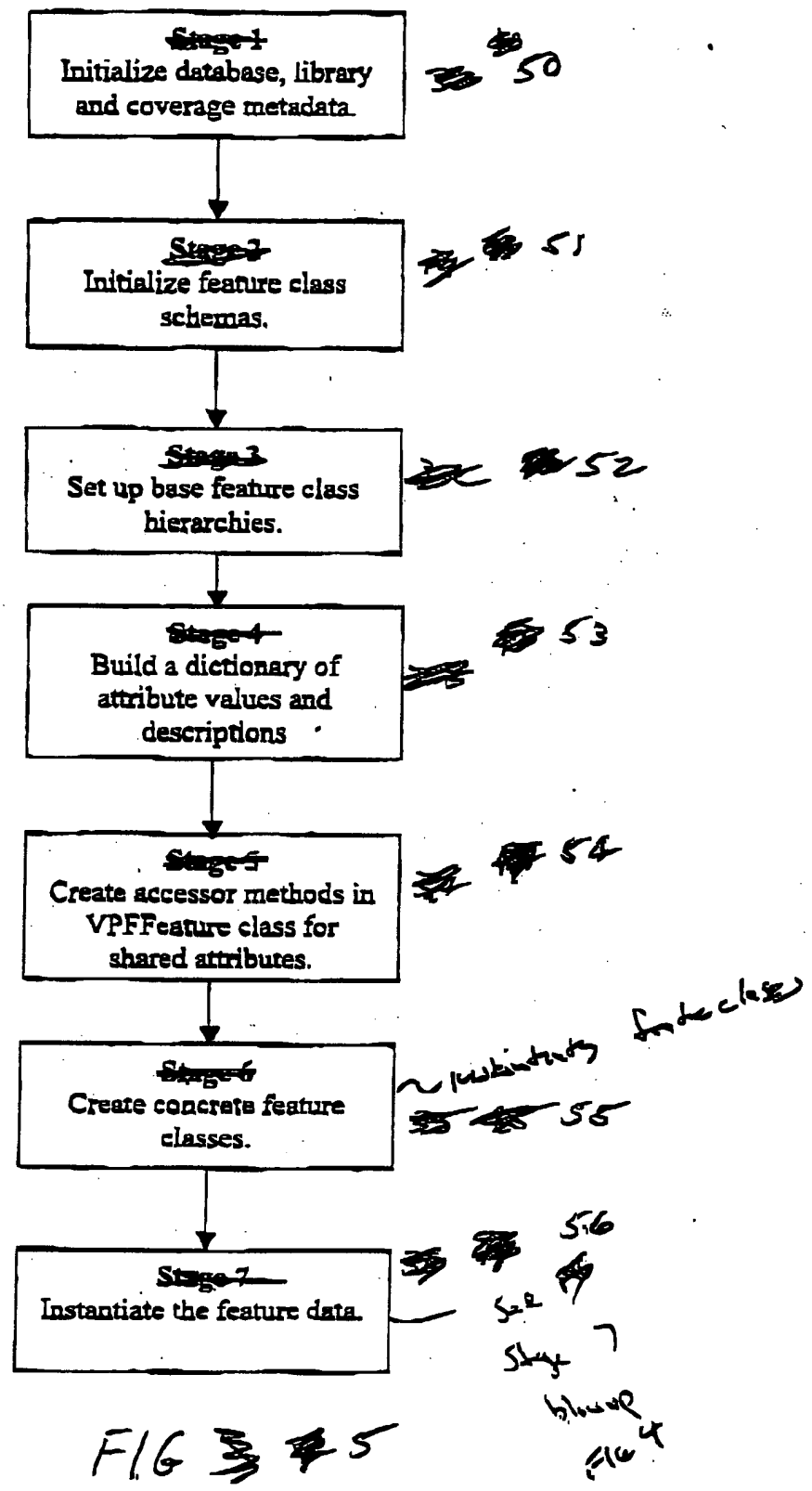
FIG 3

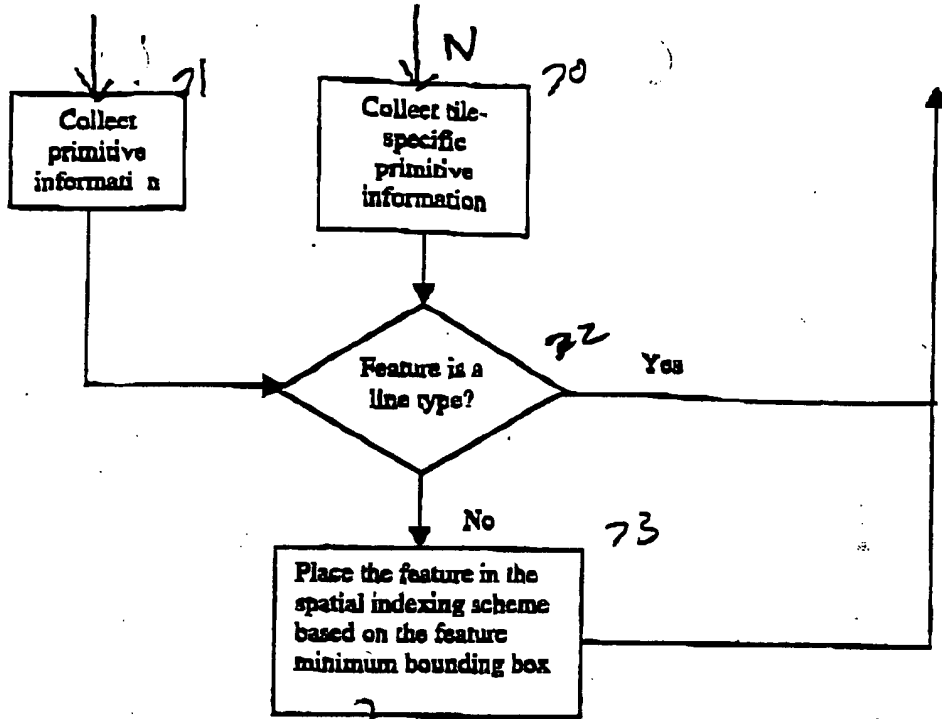
PAGE 37

Figure 1. Winged-Edge Topology

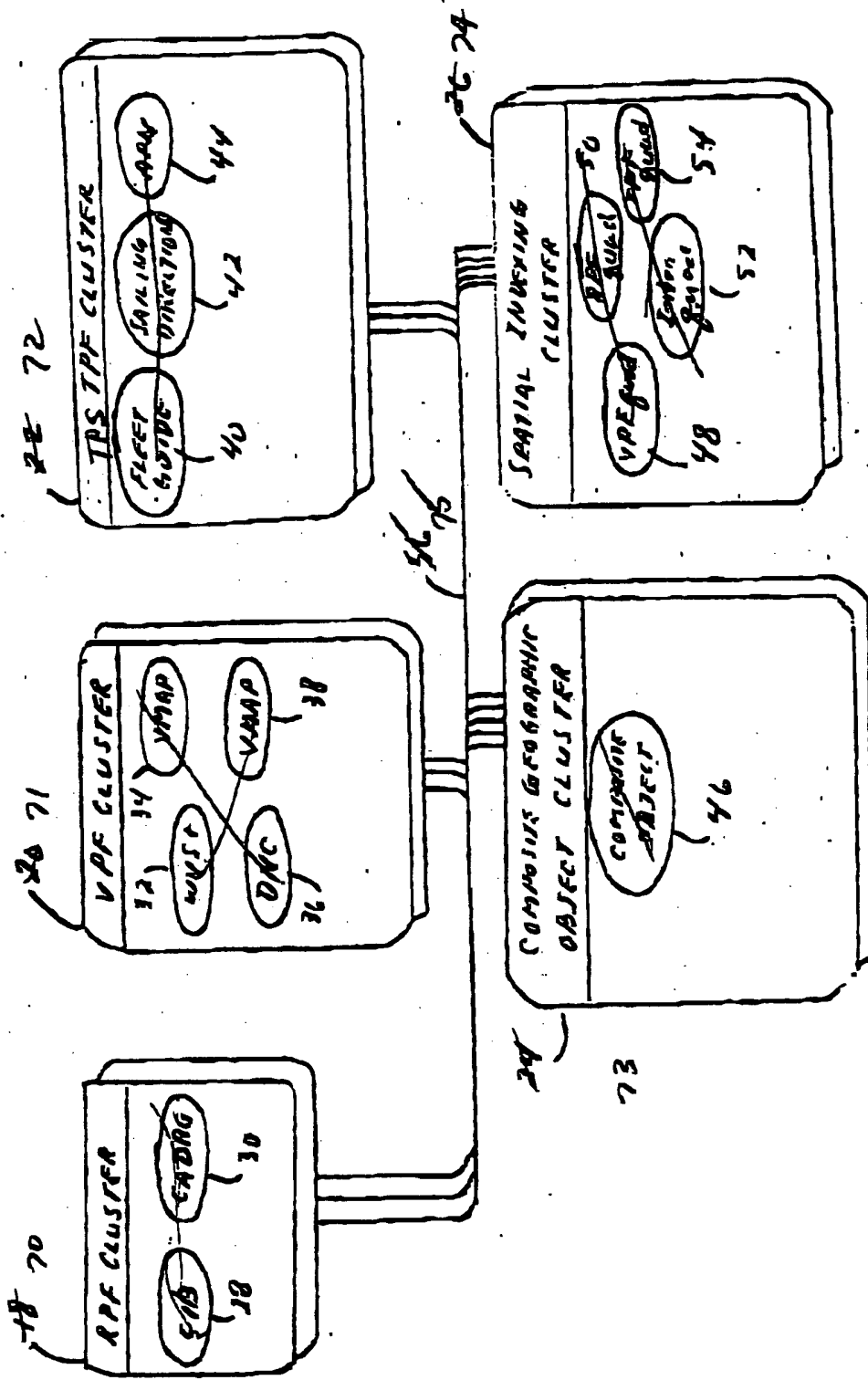
provide a mutually exclusive and colle
h partition a geographic region. In ou
efined by a set of faces and edges that do

Stages of Initialization



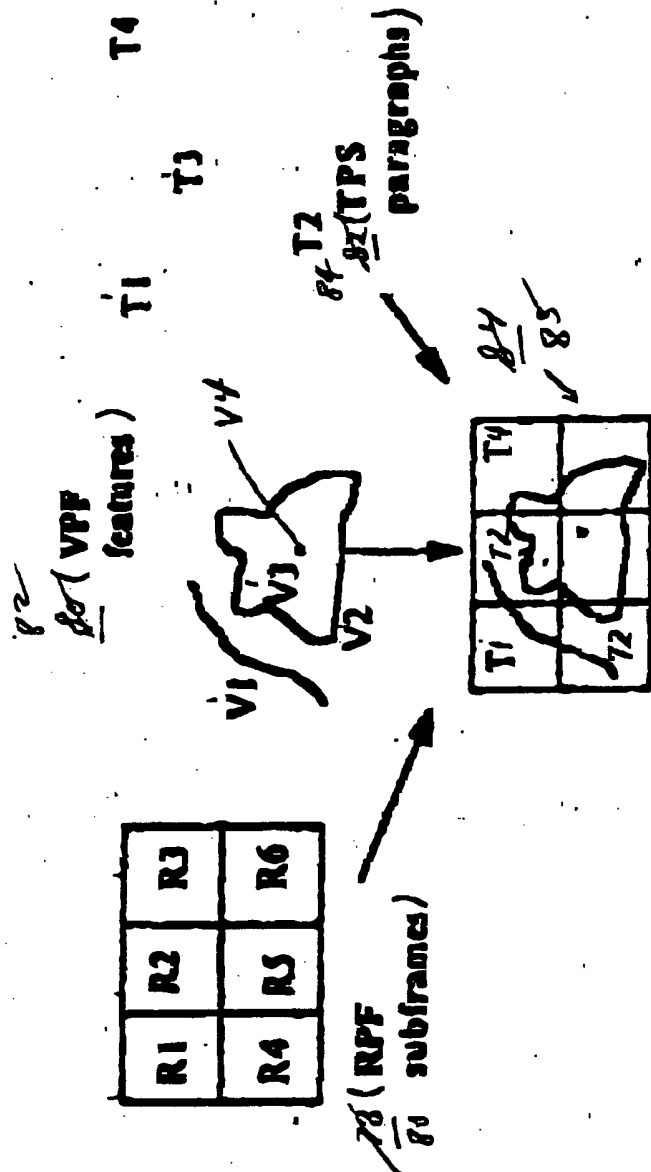


minimum boundary
provided by VPF



F16 7

76



16-8 8

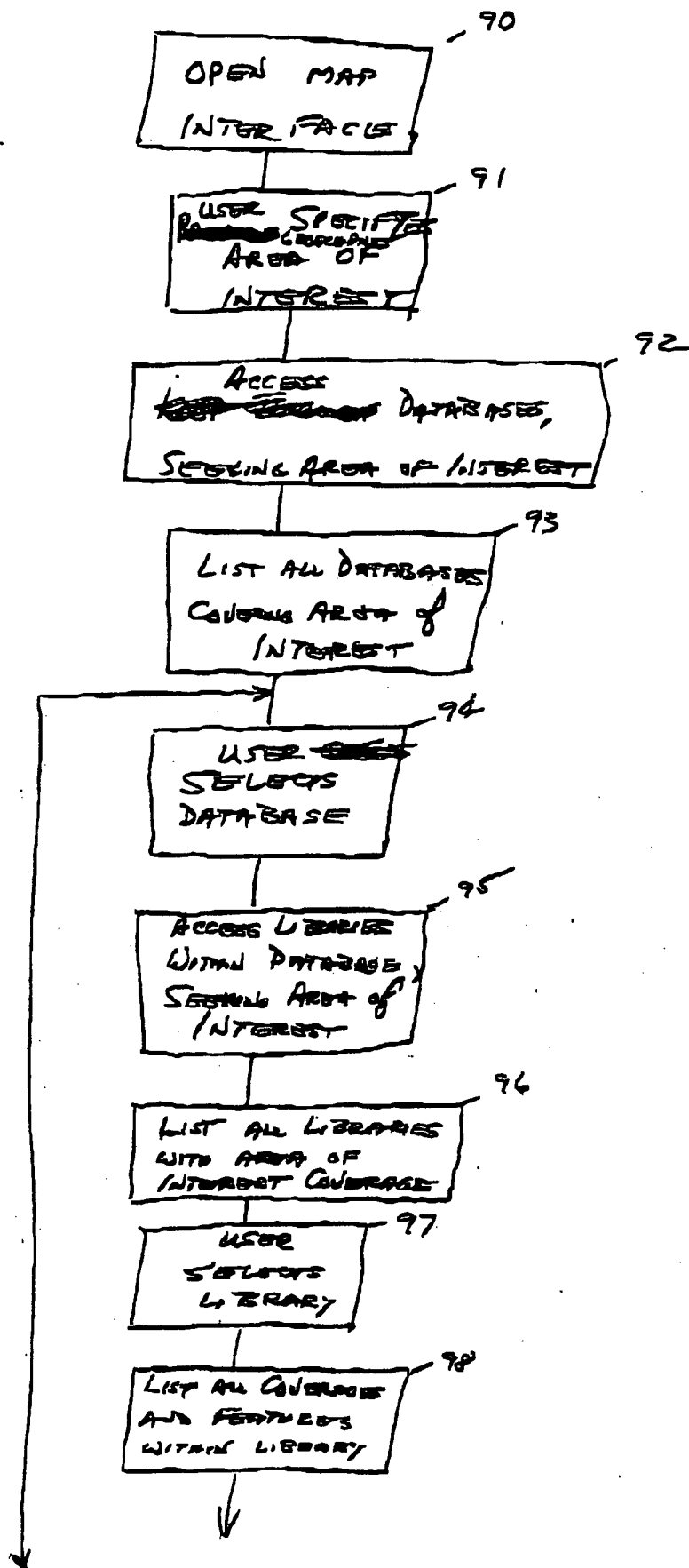


FIG. 9

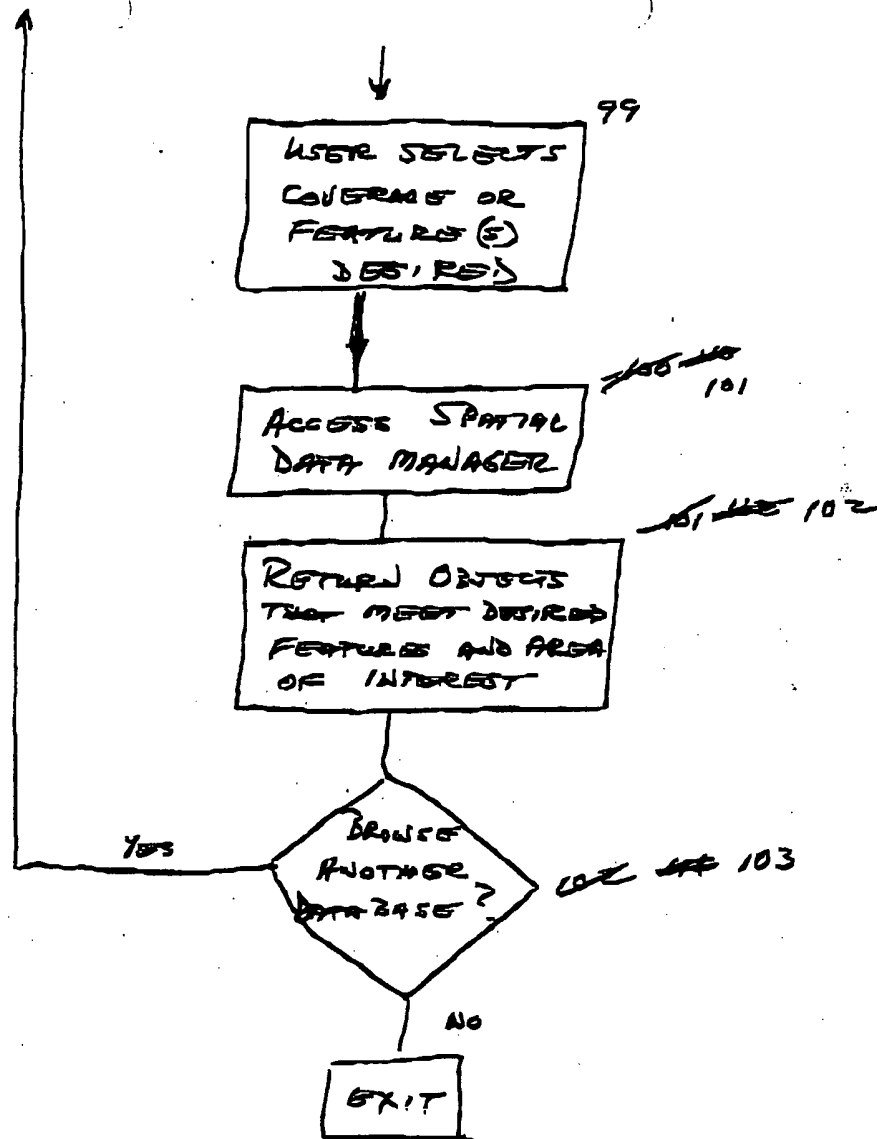


Fig 9

Such an algorithm to determine the neighboring edges can be error-prone. To prevent this possibility, a graphical user interface (GUI) has been developed to allow the user to select neighboring edges. All incident edges to the edge in consideration are displayed. With buttons for next, previous and select actions, the user has the flexibility to view and select among all possible proximate edges.

An example of an edge intersecting another edge is described below.

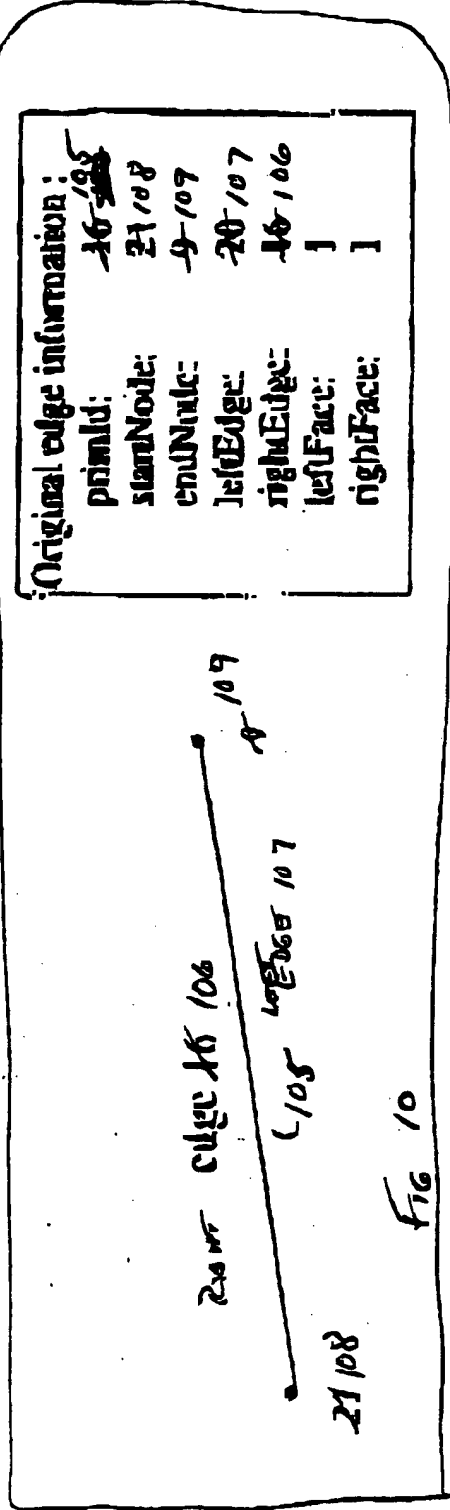


Figure 4. EdgePrim Before a New Line Feature is Added

Add a new line feature defined by three coordinate points at each vertex. Originally, this new feature is created as edge 42 with startNode and endNode of 56 and 57. The two lines intersect at two locations. Topology update should result in creation of two other edges for the new line feature and also for the intersecting line feature. The resulting winged-edge topology after adding a feature can be seen in Figure 5.

06/23/99

08:28

NRL 350 → 202

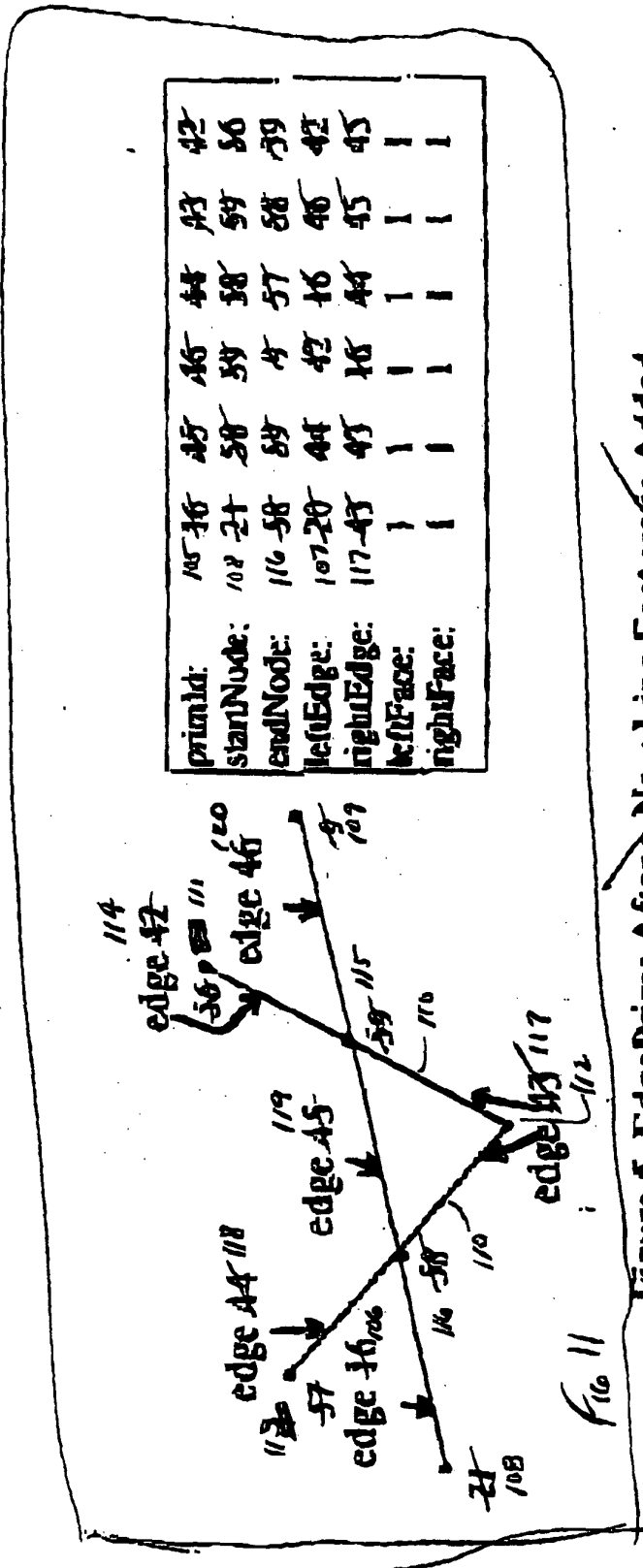


Figure 5. EdgePrims After a New Line Feature is Added

Each feature is completely defined by its primitive information and maintains its own primitive information. Since each feature's symbol has direct object pointers to its graphicElements collection of spatial primitives, updating and maintaining topology is greatly facilitated.

CONCLUSIONS AND FUTURE WORK

We have demonstrated that VPF winged-edge topology can be updated in a reasonable manner within an object-oriented framework. We have observed that

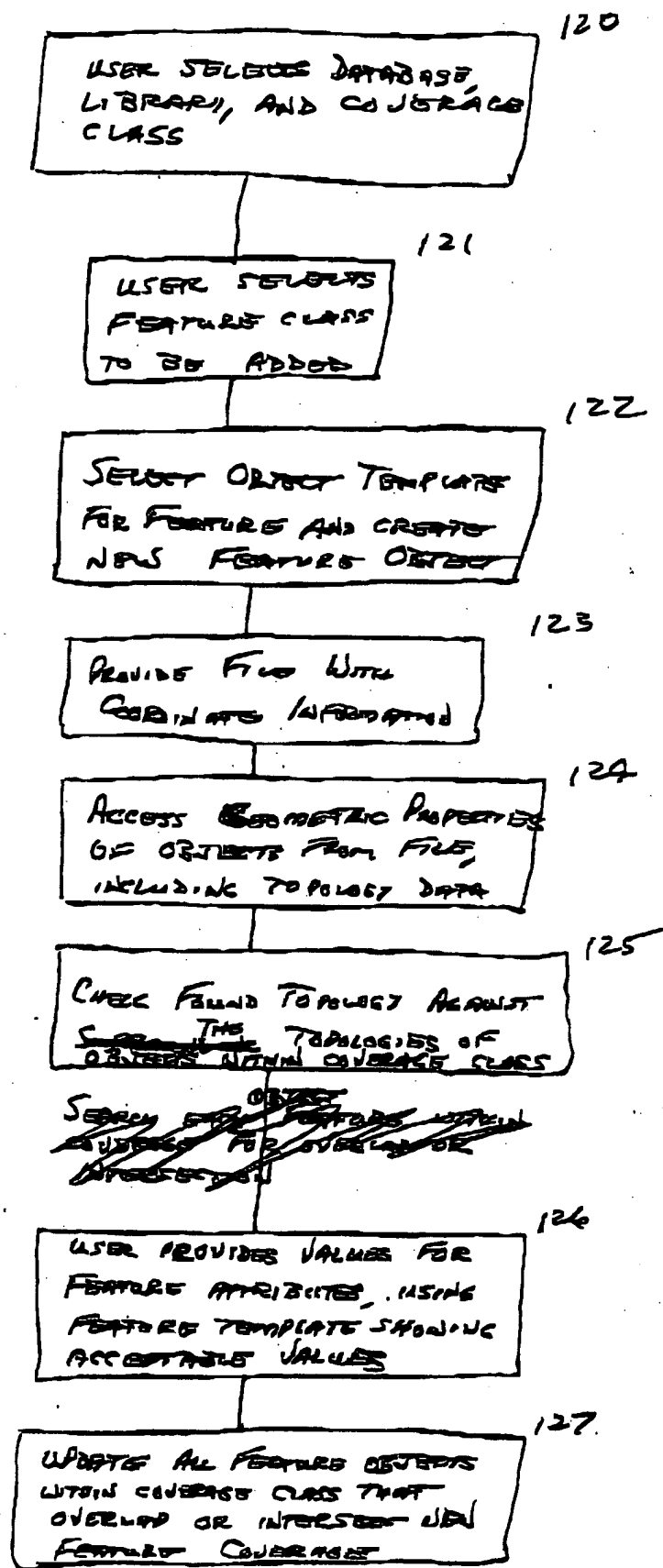


FIG 12

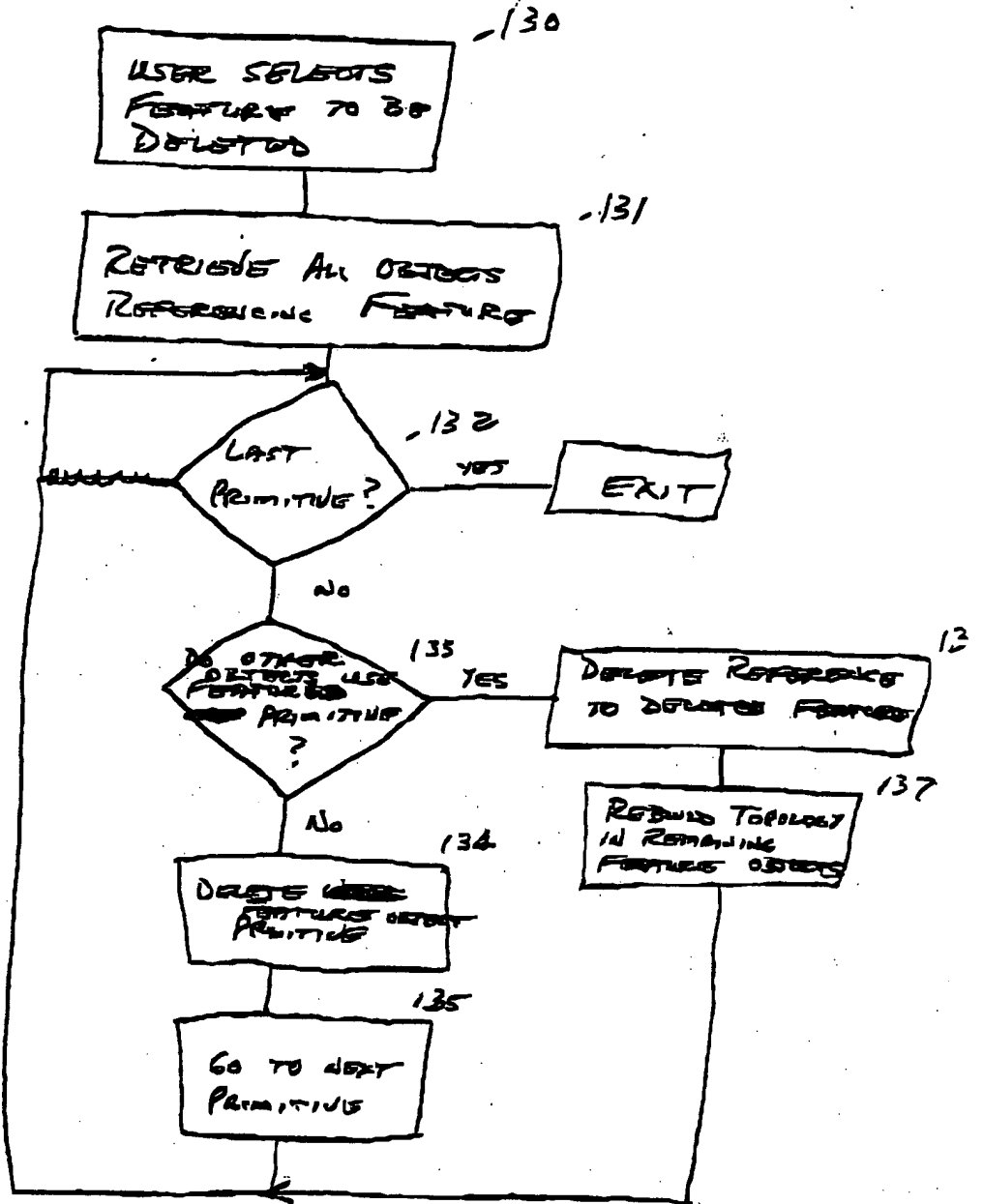


FIG 13